HumMod-Golem Edition: large scale model of integrative physiology for virtual patient simulators

Jiří Kofránek¹², Marek Mateják¹, Pavol Privitzer¹, Martin Tribula¹, Tomáš Kulhánek¹², Jan Šilar¹², Rudolf Pecinovský²

¹Charles University in Prague, Laboratory of Biocybernetics, Prague, Czech Republic
²Creative Connections, Ltd., Czech Republic

Abstract - In teaching medical decision-making, comprehensive training simulators are of great importance. These must include models of various physiological subsystems, and also integrate them into a comprehensive whole. Medical simulators have recently become a highly sought-after commercial commodity. Like an airline pilot simulator, a medical simulator is controlled by a remote operator, who manipulates the simulated patient and chooses between various scenarios to simulate different maladies. The core of a medical training simulator is a complex model of the human body’s internal physiological regulators, connected with a hardware simulator. Its detailed structure (the system of equations and the parameter values that feed into them) is usually not published, becoming a carefully-protected piece of trade secrets. There are also open source models of integrated physiological systems. One is a large model by Coleman et al. called HumMod (http://hummod.org) implemented by thousands of XML files. Our implementation of this model in the Modelica language has brought a much easier description of the simulated complex physiological relationships than XML implementation. We uncovered several mistakes in the original model, and we have modified and expanded the original model (particularly in modelling acid-base homeostasis). Our new model is called HumMod – Golem Edition (http://physiome.cz/hummod), and will provide a new theoretical basis for medical training simulator. In its implementation we will use our web simulator creation technology.

Keywords: Education, Modelica, Simulator, Virtual Patient

1 Introduction

“Tell me, I’ll forget; show me and I may remember; involve me and I’ll understand”—this ancient Chinese wisdom is also confirmed by modern learning approaches, where training simulators are widely applied.

Simulators make it possible to test the behavior of the simulated object without any risk—trying to land a virtual airplane, provide diagnostic and therapeutic interventions for a virtual patient. Or in another medical example, monitoring the behavior of individual physiological systems in response to various pathological states and therapeutic interventions.

The connection the Internet allows between interactive multimedia environment and simulation models provides quite new pedagogical opportunities, particularly when it comes to explaining complex relationships, actively exercising practical skills, and verifying theoretical knowledge. The old credo of the pioneering 17th century pedagogue John Amos Comenius—Schola Ludus, i.e. “school as a play” [4]—finds its application in incorporating multimedia educational play into training courses.

2 Virtual patient simulation

One of the most exciting innovations in the field of medical education is these virtual patient simulators [6]. Like an aircraft simulator, virtual patient simulation allows for implementing a quite new way of teaching where the student may practice diagnostic and therapeutic tasks in virtual reality, with no risk to the patient. Like an airline pilot simulator, a medical simulator is controlled by a remote operator, who manipulates the simulated patient and chooses between various scenarios to simulate different maladies. All the student’s actions are monitored, and the simulator provides material for later debriefing the diagnostic and therapeutic performance of the students [7].

New opportunities for medical education are found in virtual 3-D worlds delivered over the internet. These can include a virtual patient—a programmed avatar linked to a simulation model—and there may also be an avatar controlled by the teacher [2].

The interface of educational simulators need not be merely a computer screen. The development of haptic scanning technology and virtual reality imaging has brought a new class of simulators. These simulators are designed for training practical performance of some medical tasks (cardio-pulmonary resuscitation, catheterization, endoscopy, patient intubation, etc.) on a patient mannequin. However, hardware virtual patient simulators have also been offered in increasing complexity designed for training medical decision-making. For example, the Norwegian company Laerdal (http://www.laerdal.com/) manufactures robotic virtual patient mannequins for the training of doctors and nurses. The American company CAE Healthcare (https://caehealthcare.com) is another successful manufacturer.
whose robotic virtual patient simulators provide a highly efficient (although costly) educational aid for the training of health care professionals.

3 Scenario-driven and model-driven simulators

There are basically two approaches to managing the parameters in a virtual patient simulator.

1. Scenario-driven simulators. The behavior of these simulators is controlled by simulated disease scenarios. These scenarios are branched or statechart algorithms that respond to inputs (therapy, testing requirements, etc.), altering the parameters and showing the result. These simulators demand very complex scripts, which must be prepared by an experienced clinician. These scenarios can implement realistic results based on real patients, however, user inputs to these simulators usually consists of selecting from preset options. In these simulators it is difficult to program responses to fine-grained or quantity-based inputs (such as medication dosage, artificial ventilation settings, etc.).

2. Model-driven simulators. The behavior of these simulators is based on mathematical modeling of the physiological systems. Diseases and their treatment are simulated mainly by changing the parameters and some inputs of the model. The simulator reacts to user inputs and new values for the variables are calculated as the outputs of a mathematical model. The script requires correctly setting the model parameters for the simulated disease, which demands proper scenario debugging. On the other hand, these simulators allow entering quantified inputs (different doses of drugs, etc.). The effectiveness of this type of simulators is highly dependent on how realistic the model is. The detailed structure of these models—the system of equations and parameter values—is usually not published for commercial simulators, and becomes carefully guarded technological know-how.

4 Large-scale integrative physiology models for virtual patient simulators

Just as the theoretical foundation of an aircraft simulator is based on an airplane model, model-driven medical simulators are based on accurate models of the physiological systems in the human body. Models used as the theoretical foundation of virtual patient simulators include mathematical models not only of individual physiological subsystems, but also their interconnections, thus forming a more complex unit. The field of integrative physiology deals with the study of these connections. It seeks to describe physical reality and explain the results of experimental research, and also to create a formalized description of the how these physiological regulations are interconnected, and to explain their function in a healthy human and their misconception in the presence of various diseases.

One of the first extensive mathematical descriptions of these interconnected subsystems was published in 1972, by A. C. Guyton and two other authors [9]. From the start, the article went far beyond the scope of the physiological articles of its time. Its heart was an extensive diagram pasted in as an appendix, resembling a drawing of some electronic device. However, instead of electronic components, the diagram showed interconnected computational blocks (multipliers, dividers, summators, integrators, functional blocks, and so on that symbolized mathematical operations performed with physiological variables. Instead of writing out a system of mathematical equations, Guyton et al. used a graphical representation of mathematical relationships (Figure 1). The whole diagram was a formalized description of how the circulatory system self-regulates and its context within the body, using a graphically expressed mathematical model.

This method was quite new then. Yet the comments and reasons given for assigning the various mathematical relationships were very brief. In 1973 and 1975, further monographs [10, 11] were published providing a more detailed explanation of a number of the approaches applied.

Guyton implemented this model in Fortran. Today, designing simulation models is facilitated by specialized software environments. Matlab/Simulink by Mathworks is one. It includes a graphical simulation language, Simulink, that can be used to set up a simulation model from individual components using the mouse—providing a model of software-based simulation parts that are connected to form simulation networks. Simulink blocks highly resemble the elements used by Guyton et al. in their formalized expression of the physiological relationships; indeed, their graphic design is the only difference (Figure 2).

This similarity inspired us to resurrect the classic Guyton diagram and transform it into a functional simulation model. We tried to preserve the same external appearance of the Simulink model as in the original graphic diagram—the layout, placement of wires, variable names, and even block numbers are the same (see Figure 3).

However, simulation-based visualization of the old
diagram was not easy—namely, because there are errors in the original diagram! This is not a problem in a printed picture, but if we try to liven it up in Simulink, the model collapses immediately. For a detailed description of the errors and their corrections see [19]. Our Simulink implementation of the (corrected) Guyton model is available for download at http://www.physiome.cz/guyton. A Simulink implementation of a much more complex later Guyton model is available on this website as well. At the same time, a very detailed description is provided of all the included mathematical relationships, together with reasons for them.

In 1982 Thomas Coleman, one of Guyton’s collaborators, created a model named Human, designed primarily for educational purposes [5]. The model allowed for simulating a number of pathological conditions (cardiac and renal failure, hemorrhagic shock, etc.), and the impact of some therapeutic interventions (infusion therapy, effect of some drugs, blood transfusion, artificial pulmonary venti-

Coleman’s model was elaborated on in the large educational simulator Quantitative Circulatory Physiology (QCP). QCP can be downloaded and installed on a Windows computer. It includes a high number of variables (several thousand). The simulator allows for changing the values of approximately 750 parameters that modify physiological functions. The values of these parameters can be saved or read from an external file, which enables the user to prepare a number of scenarios for various pathological conditions. The authors of QCP have prepared many scenarios (as input files) for educational needs, and, together with appropriate comments, have made them available for free download from the QCP website. This simulator has proved useful in teaching [1].

The successor to the QCP simulator is Quantitative Human Physiology (QHP), renamed to HumMod. This simulator supports the simulation of numerous pathological conditions, including the effect of the therapy. With more than 5000 variables, HumMod seems to provide the most extensive integrated model of physiological regulations available today. Unlike QCP, whose mathematical background is hidden from the user in the C++ source code, HumMod’s [12] authors decided to separate the simulator implementation from the description of the model equations, in order to make the model structure clear for a wider scientific community.

Unlike commercial virtual patient simulators, where the structure of mathematical model is hidden, HumMod is available as open source code (the model and the simulator are available to the public at http://hummod.org).

5 HumMod Golem Edition

HumMod’s mathematical model is written in a
special XML language. The last version of HumMod (version 1.6) incorporates 4352 files spread across 1071 directories. Thanks to this fact, the model equations and their relationships are comprehensible only with difficulty, and many research teams developing medical simulators prefer to use older models as a basis for their own expansions—for example, the Guyton model of 1972 [9], or Ikeda’s models from 1979 [13]. This is the path taken, for example, by the SAPHIR (System Approach for Physiological Integration of Renal, cardiac and respiratory control) international research team, as they found the source text of the QHP/HumMod model very difficult to read and understand for the project participants [21]. Similarly, Mangourova et al. [20] recently implemented a 1992 Guyton model [3] in Simulink, rather than the more recent (but poorly legible for them) version of HumMod created by Guyton’s collaborators and students.

We were not discouraged by this difficulty, however, and have cooperated with the American authors of HumMod to improve HumMod. We have designed a special software tool [16] that creates a clear graphic representation of the mathematical relationships used, visually representing the thousands of files of source texts used by the model (Figure 4). Besides other benefits, this has also been helpful in discovering some errors in the HumMod model.

Together with American authors of HumMod we are of the opinion that source texts for the models that are the foundation of medical simulators should be publicly available, given that they are the result of freely-available theoretical studies of physiological regulations—then it becomes easy to find out to what extent the model corresponds to physiological reality. The structure of our model, which is called HumMod-Golem Edition, is published on our project website (http://physiome.cz/Hummod) in its source form, together with the definitions of all variables and equations. Unlike our American colleagues’ implementation, our model is implemented in Modelica, which makes it possible to provide a very clear expression of the model structure.

Modelica [8] is a modern simulation language. It is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems. It is often used to model mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. Unlike other object-oriented languages, classes in Modelica may contain equations. Each class in Modelica can be externally represented by user defined icon. A component in Modelica therefore represents an instance of class for which equations or parameters are defined. Components (represented as icons) can be linked through connectors. The user graphically links these icons to create a system of equations. The structure of the model in Modelica therefore reflects the structure of the modeled system, unlike the model in Simulink, which expresses the structure of the calculation procedure rather than the structure of the modeled reality.

Unlike the block-oriented simulation environment in Simulink, the structure of Modelica models corresponds to the physical essence of the modeled reality (the compiler takes care of the “dirty work” of solving the resulting system of algebraic differential equations). Models in Modelica are, compared to those in Simulink, clearer and more self-documenting. This advantage can be demonstrated, for example, by comparing an implementation of the classic
Figure 7: Structure of the cardiovascular component (CVS class from Fig. 6).

Figure 8: Structure of the systemic circulation component (SystemicCirculation class from Figure 7).

Figure 9: Structure of the systemic peripheral circulation component (Peripheral class from Figure 8).

Figure 10: Structure of the splanchnic circulation component (SplanchnicCirculation class from Figure 9).

Figure 11: Structure of the gastrointestinal vascular resistance component (GITract class from Figure 10).

Figure 12: Structure of a component calculating the influence of alpha receptor stimulation on gastrointestinal vascular resistance (AlphaReceptors class from Figure 11).
Guyton model [9] in Simulink (Figure 3) and in Modelica (Figure 5)

The HumMod model has been modified and expanded, particularly in the field of blood gas transfer and regulation of acid-base homeostasis. Our goal is to create an educational simulator designed primarily for teaching emergency medicine, in which disorders of blood-gases and acid-base homeostasis occur frequently. Among other sources, our modifications stemmed from our earlier complex model of physiological regulations, the core of the educational simulator Golem [14].

See Figures 6–12 for illustrations of the hierarchic structure used by Hummod-Golem Edition.

The model allows a user to simulate a number of physiological and pathophysiological actions—for instance, the failure of individual organs and organ systems, and the body’s subsequent adaptation; the effect of any chosen therapy; response to physical load; and the body’s response to a change of some external condition (for example, a rise in temperature). Hummod-Golem Edition provides a theoretical foundation, and is used by the medical educational simulator BodyLight. However, its further development and identification are only the first challenges that must be faced. Another problem consists in programming the simulator itself as an educational aid. Our aim is to make the simulator available as a teaching aid through the Internet. Our web simulator design technology [15] (which is described in Figure 13) will be used in its design.

Instruction models (and apparently not only complex ones with hundreds of variables) in themselves therefore are not enough for efficient use in teaching. They must be accompanied by explanation of their application—using interactive educational applications at best. The possibility of using all advantages of virtual reality to explain complex pathophysiological processes arises only upon establishing connection between explanation and interactive simulation. In order to link the possibilities offered by interactive multimedia and simulation models in medical teaching, we have designed the concept of an Internet computer project, the Atlas of Physiology and Pathophysiology [17, 18], conceived as a multimedia instruction aid that should help to explain, in a visual way using the Internet and simulation models, the function of individual physiological subsystems, the causes and manifestations of their disorders – see http://physiome.cz/ atlas. The Atlas thus combines explanation (using audio and animation) with interactive simulation play with physiological subsystems models, all available for free from the Internet.

6 From art to industry in designing of virtual patient simulators

Individual enthusiasts created the first educational programs at the turn of the 80s, excited by the potential of personal computers. Their time is long gone. Today, high-quality educational software must utilize the potential of new information and communication technologies. This means it must be built on more than the diligence and enthusiasm of individuals. It is a demanding and complicated process, requiring a creative team of specialists across various professions: Experienced teachers whose scenarios provide the foundation of a good-quality educational application; system analysts who create the simulation models themselves; artists and UI designers who create the external form and software interface; and finally, information science specialists (programmers) who “stitch up” the whole application into its final form [15].

For such interdisciplinary cooperation to be efficient, numerous development tools and methodologies are needed for every stage of development; such tools and methodologies make the work of individual team members easier and help them to overcome interdisciplinary barriers. Considerable efforts must be devoted to the process of creating and mastering the tools, but it pays off in the end. With Golem/BodyLight, just such a cross-disciplinary team was able to unify excellent underlying code (from HumMod) with new visual and analytical clarity inspired by the Guyton model, to create a system which is more than the sum of its parts.

7 Conclusions

Complex hierarchical models of human integrative physiology are the key to model-driven virtual patient simulators. Such simulators include models of not only individual physiological subsystems, but also of their connection into more complex units. Modelica is a very convenient developing tool for designing these models.

The HumMod model clearly shows the benefits using the Modelica language. If we compare the complex structure of the original HumMod model written in XML [11] with implementations done in Modelica (Figures 6–12), we can clearly see that the implementation done in Modelica creates a transparent and legible model structure and therefore offers easier model modifications. Our Modelica implementation of a modified and extended
HumMod model is the key part of developing the BodyLight web simulator for medical education.

8 References


Acknowledgement

This paper describes the outcome of research that has been accomplished as part of research program funded by the Ministry of Industry and Trade of the Czech Republic by the grant FR—TI3/869 and by The Ministry of Education, Youth and Sports by the grant SVV-2013-266509.

email to corresponding author: kofranek@gmail.com